A Stochastic Algorithm for Probabilistic Independent Component Analysis

Stéphanie Allassonniere, Laurent Younes

A Stochastic Algorithm for Probabilistic Independent Component Analysis

Stéphanie Allassonnière[??], **and Laurent Younes**[??]

Centre de Mathematiques Appliquées
Ecole Polytechnique
Route de Saclay
91128 Palaiseau, FRANCE
e-mail: Stephanie.Allassonniere@polytechnique.edu

Center for Imaging Science
Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218
e-mail: Laurent.Younes@jhu.edu

Abstract: The decomposition of a sample of images on a relevant subspace is a recurrent problem in many different fields from Computer Vision to medical image analysis. We propose in this paper a new learning principle and implementation of the generative decomposition model generally known as noisy ICA (for independent component analysis) based on the SAEM algorithm, which is a versatile stochastic approximation of the standard EM algorithm. We demonstrate the applicability of the method on a large range of decomposition models and illustrate the developments with experimental results on various data sets.

AMS 2000 subject classifications: Primary 60J22; secondary 62F10,62M40.
Keywords and phrases: Independent component analysis; Independent factor analysis; Stochastic approximation; EM algorithm; Statistical modelling; Image analysis., LaTeX 2_ε.

1. Introduction

Independent Component Analysis (ICA) is a statistical technique which aims at representing a data set of random vectors as linear combinations of a fixed family of vectors with statistically independent coefficients. It has found numerous applications, starting with source separation [1], for which it has been designed initially, but also including image analysis and more generally any situation in which a decomposition of a large set of variables into simple components is needed. It has proved to provide representations that are qualitatively very different from, say, principal component analysis (PCA) [2].

One of the drawbacks of ICA is that it does not come (like PCA does) with a natural selection method for the most important components. In the original formulation, the number of independent components is equal to the dimension of the variables, so that the decomposition is achieved without dimensional reduction. When a limited amount of data is available, the validity of this decomposition is generally subject to caution due to over-fitting. Probabilistic ICA, in which part of the signal is modelled as noise, provides an interesting approach to this issue.

ICA and Probabilistic ICA admit formulations in terms of generative models approximating the distribution of the data, allowing for the use of well-understood statistical methods for training and validation. ICA, for example, represents an observed d-dimensional random variable, $\boldsymbol{X}$, as

$$\boldsymbol{X} = \sum_{j=1}^{d} \beta^j \boldsymbol{a}_j \,, \tag{1.1}$$

where $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_d) \in \mathbb{R}^{d \times d}$ are parameters (called decomposition vectors) and $\beta^1, \ldots, \beta^d$ are independent scalar random variables. This model can be specialised by specifying the distribution

*Laurent Younes's research was partially supported by NSF ITR:0427223.

of the β^j's and basic estimation procedures can be used, like maximum likelihood (or maximum entropy) as in [3].

As we remarked, estimating a full d by d matrix of parameters (called decomposition matrix) may be unrealistic when the number of observations is limited, and one may prefer using a Probabilistic ICA model, given by

$$\boldsymbol{X} = \sum_{j=1}^{p} \beta^j \boldsymbol{a}_j + \sigma \boldsymbol{\varepsilon}\,, \tag{1.2}$$

where $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p) \in \mathbb{R}^{d\times p}$ now represent $d \times p$ parameters, $\beta^1, \ldots, \beta^p$ are independent scalar random variables and $\boldsymbol{\varepsilon}$, the noise, follows a standard normal distribution (we here take the standard deviation, σ to be a fixed scalar, also a parameter). Such models can also address the fact that for many types of data, only a small number of components is required to describe an input vector on average.

Training (Probabilistic) ICA requires to estimate the decomposition matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p)$ (and the noise variance in the probabilistic case) based on observations $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$. With model (1.1), the independent components $\beta^1, \ldots, \beta^d$ can be computed from $\boldsymbol{X}$ by inverting $\boldsymbol{A}$ and most of the learning methods estimate $\boldsymbol{W} = \boldsymbol{A}^{-1}$ by ensuring that $\boldsymbol{W}\boldsymbol{X}$ has independent components. With Probabilistic ICA, this is obviously not possible; the d-dimensional vector $\boldsymbol{X}$ is modelled as a function of the $(p+d)$-dimensional variable $(\boldsymbol{\beta}, \boldsymbol{\varepsilon})$ and we have partial observations.

A possible approach is to first implement some dimension reduction (typically PCA) to the data before applying standard ICA to the projected components [4, 5]. But training probabilistic ICA according to the statistical model that it actually describes is certainly a more satisfactory approach. In this setting, the adopted solution in the literature is most of the time to maximise the likelihood for the joint distribution of $\boldsymbol{X}$ and $\boldsymbol{\beta}$, simultaneously in the parameters and in the unobserved variables [6]. This therefore attempts to solve the parametric estimation and reconstruction problems at the same time. However, the estimation of both $\boldsymbol{X}$ and $\boldsymbol{\beta}$ may be a risky procedure, often inducing biased estimators. As we will show in our experiments, these approaches have good results when the noise level is small (as already noticed in [7]), but these results can significantly degrade otherwise (see Section 5, or [8] for a similar observation made in a different context).

In this paper, we estimate the parameters by maximum likelihood of the *observed variables*, therefore averaging out the unobserved $\boldsymbol{\beta}$. The reconstruction problem (estimating $\boldsymbol{\beta}$ from $\boldsymbol{X}$), which is important, for example to define efficient lossy compression methods, can then be solved using the estimated parameters. These are two separate problems.

So our focus will be on maximum likelihood estimation of the parameters ($\boldsymbol{A}$, σ and some additional parameters describing the distribution of the $\boldsymbol{\beta}$'s), based on partial observations. The Expectation - Maximisation (EM) algorithm is the most commonly used method for this purpose. However, it is intractable in our case because of the difficulty to compute conditional expectations given the observations, which are needed in the E-step. We will rely on a stochastic approximation to the EM (called SAEM [9, 10]) which only requires being able to sample from this conditional distribution in order to provide converging results. This algorithm compensates the larger convergence time generally associated to stochastic approximations by much simpler iteration steps, since sampling hidden variables is most of the time far easier than computing conditional expectations. Moreover the construction applies to many different probabilistic distributions. This means that there are almost no restriction to the range of statistical models that can be used for the unobserved independent variables.

To illustrate this, the paper will describe a series of models and variants that lead to various instances of probabilistic ICA, all leading to fairly similar learning algorithms. We introduce these models in Section 2. The parametric estimation method, including the SAEM algorithm, is described in Section 3 and the reconstruction of hidden variables is discussed in Section 4. Experimental results with both synthetic and real data are presented in Section 5.

2. Models

We start with some general assumptions on the data, that will be made specific in the experiments. We assume the observation is a set of vectors which take values in $\mathbb{R}^d$. Let $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ be the training observations, which are assumed to be independent and identically distributed. We will denote by $\boldsymbol{X}$ a generic variable having the same distribution as the $\boldsymbol{X}_k$'s. The jth coordinate of $\boldsymbol{X}$ (resp. $\boldsymbol{X}_k$) will be denoted X^j (resp. X_k^j).

We assume that $\boldsymbol{X}$ can be generated in the form

$$\boldsymbol{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} \beta^j \boldsymbol{a}_j + \sigma \boldsymbol{\varepsilon}\,, \tag{2.1}$$

where $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, $\boldsymbol{a}_j \in \mathbb{R}^d$ for all $j \in \{1, ..., p\}$, $\boldsymbol{\varepsilon}$ is a standard d dimensional Gaussian variable and $\beta^1, \ldots, \beta^p$ are p independent scalar variables, the distribution of which being specified later. Let $\boldsymbol{\beta}$ denote the p-dimensional variable $\boldsymbol{\beta} = (\beta^1, \ldots, \beta^p)$. To each observation $\boldsymbol{X}_k$ is therefore associated hidden realisations of $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$, which will be denoted $\boldsymbol{\beta}_k$ and $\boldsymbol{\varepsilon}_k$.

Denote $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p)$. It is a d by p matrix and one of the parameters of the model. Another parameter is σ, which will be a scalar in our case (a diagonal matrix being also possible). Additional parameters will appear in specific models of $\boldsymbol{\beta}$ which are described in the following subsections. In some of these models, it will be convenient to build $\boldsymbol{\beta}$ as a function of new hidden variables, which will be denoted $\boldsymbol{Z}$.

The models that we describe are all identifiable, as proved in [11], with the obvious restriction that $\boldsymbol{A}$ is identifiable up to a permutation of its columns. When the distribution of $\boldsymbol{\beta}$ is symmetrical, the columns $\boldsymbol{A}$ are also identifiable up to a sign change.

2.1. Logistic distribution (Log-ICA)

We start with one of the most popular models, in which each β^j follows a logistic distribution with fixed parameter 1/2. The associated cumulative distribution function is $P(\beta^j \leq t) = 1/(1 + \exp(-2t))$.

For this model, the parameters to estimate are $\theta = (\boldsymbol{A}, \sigma^2, \mu_0)$. Hidden variables are $\boldsymbol{Z} = \boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$. This is the model introduced in the original paper of Bell and Sejnowsky [3], and probably one of the most commonly used parametric model for ICA.

2.2. Laplacian distribution (Lap-ICA)

A simple variant is to take β^j to be Laplacian with density $e^{-|t|}/2$. The parameter still is $\theta = (\boldsymbol{A}, \sigma^2, \boldsymbol{\mu}_0)$. Hidden variables are $\boldsymbol{Z} = \boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$.

2.3. Independent Factor Analysis (IFA)

The IFA [12, 13] model is a special case of probabilistic ICA in which the distribution of each coordinate β^j is assumed to be a mixture of Gaussians. To allow for comparison with the state of the art, we will test this model as well. We will here use a restricted definition of the IFA model which will be consistent with the other distributions that we are considering in this paper, ensuring that the β^j's are independent with identical distribution, and that this distribution is symmetrical.

More precisely, we will introduce two new hidden variables, the first one (representing the class in the mixture model), being denoted $(t^1, \ldots, t^p)$ and the second one is a random sign change, $(b^1, \ldots, b^p)$ for each component. Each t^j takes values in the finite set $\{0, 1, \ldots, K\}$, with respective probabilities $w_0, \ldots, w_K$, and b^j takes values ± 1 with probability $\frac{1}{2}$. Then let

$$\beta^j = b^j \sum_{k=1}^{p} m_k \delta_k(t^j) + Y^j$$

where Y^j is standard Gaussian. In other terms, β^j is a mixture of $2K+1$ Gaussians with unit variance, the first one being centred, and the following ones having means $m_1, -m_1, m_2, -m_2, \ldots$.

The parameters of this model are therefore $\theta = (A, \sigma^2, (w_k, m_k)_{1 \leq k \leq K})$. Hidden variables are $\boldsymbol{Z} = (\boldsymbol{\beta}, \boldsymbol{b}, \boldsymbol{t})$. Note that, even if we use a simplified and symmetrized version of the model originally presented in [12], the stochastic approximation learning algorithm that will be designed in Section 3.2 immediately extends to the general case where the means depend on the index j.

2.4. Exponentially scaled Gaussian ICA (EG-ICA)

In this model, we let $\beta^j = s^j Y^j$ where $\boldsymbol{Y}$ is a standard Gaussian vector, $s^1, \ldots, s^p$ are independent exponential random variables with parameter 1, also independent from Y and $\boldsymbol{\varepsilon}$. In this case, we can write

$$\boldsymbol{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} s^j Y^j \boldsymbol{a}_j + \boldsymbol{\sigma}\epsilon. \tag{2.2}$$

Hidden variables are $\boldsymbol{Z} = (\boldsymbol{s}, \boldsymbol{Y})$ and $\boldsymbol{\varepsilon}$, and the parameter is $\theta = (\boldsymbol{A}, \sigma^2, \boldsymbol{\mu}_0)$.

It's not too hard to prove (see the Appendix) that this model is such that $\log[P(\beta^i > t)]$ is asymptotically proportional to $(-t^{2/3})$, providing sub-exponential tails.

2.5. Bernoulli-censored Gaussian (BG-ICA)

With some types of data, only a sub-group of all the decomposition vectors is required to describe one input vector. In contrast with the logistic or Laplacian models for which coefficients vanish with probability zero, we now introduce a discrete switch which "turns them off" with positive probability. Here, we model the hidden variables as a Gaussian-distributed scale factor multiplied by a Bernoulli random variable. We therefore define $\beta^j = b^j Y^j$, using the same definition for $\boldsymbol{Y}$ as in section 2.4 and letting b^j have a Bernoulli distribution with parameter $\alpha = P(b^j = 1)$. We assume that all variables $b^1, \ldots, b^p, Y^1, \ldots, Y^p, \varepsilon$ are independent. The complete model for X has the same structure as before, namely

$$X = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} b^j Y^j \boldsymbol{a}_j + \sigma\boldsymbol{\varepsilon}. \tag{2.3}$$

Parameters in this case are $\theta = (\boldsymbol{A}, \sigma^2, \alpha, \boldsymbol{\mu}_0)$ and hidden variables are $\boldsymbol{Z} = (\boldsymbol{b}, \boldsymbol{Y})$ and $\boldsymbol{\varepsilon}$. Using a censoring distribution in the decomposition is a very simple way to enforce sparsity in the resulting model.

2.6. Exponentially scaled Bernoulli-censored Gaussian (EBG-ICA)

We can combine the two previous models, using both the censoring variable and the exponential scale to benefit from both model advantages. The complete model for X is

$$X = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} s^j b^j Y^j \boldsymbol{a}_j + \sigma\boldsymbol{\varepsilon}. \tag{2.4}$$

Since the exponential law has fixed variance, the parameters of interest are the same as in the BG-ICA model, i.e. $\theta = (\boldsymbol{A}, \sigma^2, \alpha, \boldsymbol{\mu}_0)$. The hidden variables are $\boldsymbol{Z} = (\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{Y})$ and $\boldsymbol{\varepsilon}$.

2.7. Exponentially-scaled ternary distribution (ET-ICA)

The previous models include an switch which controls whether the component is active in the observation or not. One may want to have either an activation or an inhibition of the corresponding

decomposition vector. To this purpose, we introduce a discrete model for $\boldsymbol{Y}$, each component taking only values -1, 0 or 1. We define $\beta^j = s^j Y^j$, where $s^1, \ldots, s^p$ are i.i.d. exponential variables with parameter 1. We let $\gamma = P(Y^j = -1) = P(Y^j = 1)$, providing a symmetric distribution for the components of $\boldsymbol{Y}$. As before, all hidden variables are assumed to be independent. The model is

$$X = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} s^j Y^j \boldsymbol{a}_j + \sigma \boldsymbol{\varepsilon}. \tag{2.5}$$

Hidden variables here are $\boldsymbol{Z} = (\boldsymbol{s}, \boldsymbol{Y})$ and $\boldsymbol{\varepsilon}$, the parameter being $\theta = (\boldsymbol{A}, \sigma^2, \gamma, \boldsymbol{\mu}_0)$.

The interpretation of the decomposition is that each component has a fixed effect, up to scale, which can be positive, negative or null. The model can therefore be seen as a variation of the Bernoulli-Gaussian where the effect can be a weighted inhibitor as well as a weighted activator. This allows selective appearance of decomposition vectors and therefore refine the characterisation of the population.

2.8. Single-scale ternary distribution (TE-ICA)

The previous model can be simplified by assuming that the exponential scale is shared by all the components, i.e., we let $\beta^j = sY^j$ where s is exponential with parameter 1, and Y^j has the same ternary distribution as in the ET-ICA model. The decomposition now is

$$X = \boldsymbol{\mu}_0 + s \sum_{j=1}^{p} Y^j \boldsymbol{a}_j + \sigma \epsilon. \tag{2.6}$$

Hidden variables here are $(s, \boldsymbol{Y})$, the parameter being $\theta = (\boldsymbol{A}, \sigma^2, \gamma, \boldsymbol{\mu}_0)$. Notice that this model is not explicitly an ICA decomposition, since the components are only independent given the scale. Notice also that we assume that the scaling effect acts on the components, not on the observation noise which remains unchanged.

Probabilistic-ICA in general is obviously a very efficient representation for lossy compression of random variables, since, if the noise is neglected, and as soon as the parameters $\boldsymbol{\mu}_0$ and A are known, one only needs to know the realisation of $\boldsymbol{\beta}$ (hopefully with $p << d$), to reconstruct an approximation to the signal. In the present model, the transmission of $\boldsymbol{\beta}$ only requires sending the scalar scale, s, and p ternary variables. If many components vanish (i.e., if γ is significantly smaller than $1/2$), compression is obviously even more efficient.

In this model (and for the previous two also), the sparsity of the representation will obviously depend on the number of selected components, p, that we suppose given here. When p is too small, it is likely that the model will find that censoring does not help and find $\gamma = 1/2$ (or $\alpha = 1$ in the Bernoulli-Gaussian model). Adding more components in the model generally results in α and γ decreasing, enabling some components to be switched off. This effect is illustrated in Section 5.

Finally, let's remark that, although the results in [11] do not directly apply to this model (the components are not independent, since they share the same scale factor), they can be applied to the conditional distribution given the scale to prove identifiability (since the scale distribution is fixed).

2.9. Playing with the average

Clearly, all the previous models admit a centred submodel in which $\boldsymbol{\mu}_0 = 0$, which can be preferred in some cases. In this case ($\boldsymbol{\mu}_0 = 0$), it may be interesting to allow for some shift in the distribution of the components, replacing β^j by $\mu + \beta^j$ where μ is a one-dimensional parameter. This is therefore equivalent to to modelling $\boldsymbol{\mu}_0 = A\boldsymbol{\mu}$ where μ is a p-dimensional vector with all coordinates equal to μ. When dealing with scaled, or censored models, one can decide to apply the shift before or after censoring or scaling. For example, one can define a shifted Bernoulli-Gaussian model by replacing Y^i by $\mu + Y^j$ in Section 2.5, which results in shifting β^j only when it is not censored.

Another choice that can also be interesting is to model the signal with a random, scalar, offset (or AC component). One way to achieve this is to impose that one of the columns of the matrix $\boldsymbol{A}$ is the d-dimensional vector $(1,\dots,1)^T$. In this case, it is natural to separate the distribution of the offset coefficient from the ones of other components, as customary in compression (the offset coefficient should not be censored, for example). A simple choice is to provide it with a logistic or Laplacian distribution. This is illustrated in the next model.

2.10. Single-scale ternary distribution with offset (TEoff-ICA)

In this model, the mean $\boldsymbol{\mu}_0$ is not a parameter and is not the same for all the observed vectors, so that this random effect (in opposition to the fixed effect it had in the previous models) now is a hidden variable. We furthermore assume that this random variable, denoted $\boldsymbol{\mu}$ takes the form $\boldsymbol{\mu} = (\mu, ..., \mu) \in \mathbb{R}^d$ where μ is Laplacian. So μ can be interpreted as an offset acting simultaneously on all coordinates of $\mathbf{X}$. This yields the following model:

$$X = \boldsymbol{\mu} + s\sum_{j=1}^{p} Y^j \boldsymbol{a}_j + \sigma\varepsilon\,, \tag{2.7}$$

where s follows an exponential distribution with parameter 1 and Y^j are ternary variables with $\gamma = P(Y^j = -1) = P(Y^j = 1)$. The hidden variables are $(s, \boldsymbol{Y}, \mu)$ and the parameters (A, σ^2, γ).

3. Maximum likelihood estimation

3.1. Notation

The previous models are all built using simple generative relations $\boldsymbol{Z} \to \boldsymbol{\beta}$ and $(\boldsymbol{\beta}, \boldsymbol{\varepsilon}) \to \boldsymbol{X}$. Our goal here is to estimate the parameters that maximise the likelihood of the observation of n independent samples of $\boldsymbol{X}$ that we will denote $\boldsymbol{x}^{*n} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$.

Let $q_m(\boldsymbol{z};\theta)$ denote the prior likelihood of the hidden (or missing) variable $\boldsymbol{Z}$ that generates $\boldsymbol{\beta}$. Denote $q_c(\boldsymbol{x}|\boldsymbol{z};\theta)$ the conditional distribution of $\boldsymbol{X}$ given $\boldsymbol{Z} = \boldsymbol{z}$ which is, in all our models, a Gaussian distribution centred at the ICA decomposition. The joint density is

$$q(\boldsymbol{x}, \boldsymbol{z};\theta) = q_c(\boldsymbol{x}|\boldsymbol{z};\theta) q_m(\boldsymbol{z};\theta)$$

and the marginal distribution of $\boldsymbol{X}$ has density

$$q_{obs}(\boldsymbol{x};\theta) = \int q_c(\boldsymbol{x}|\boldsymbol{z};\theta) q_m(\boldsymbol{z};\theta) d\boldsymbol{z}.$$

Our goal is to maximise the likelihood of the observations, namely to find

$$\hat{\theta}_n = \operatorname*{argmax}_{\theta} q^{*n}_{obs}(\boldsymbol{x}^{*n};\theta) \text{ with } q^{*n}_{obs}(\boldsymbol{x}^{*n};\theta) = \prod_{k=1}^{n} q_{obs}(\boldsymbol{x}_k;\theta). \tag{3.1}$$

3.2. SAEM Algorithm

This problem can, in principle, be solved using the Expectation Maximisation (EM) algorithm. With the EM, a local maximum of the likelihood is computed recursively while replacing the missing variables with a conditional expectation. For each observation $\boldsymbol{x}_k$ and parameter θ, we define the conditional density of $\boldsymbol{Z}$ by

$$\nu_{k,\theta}(\boldsymbol{z}) = q(\boldsymbol{z}|\boldsymbol{X} = \boldsymbol{x}_k;\theta)\,. \tag{3.2}$$

The EM algorithm iterates the following two steps, where t indexes the current iteration;

E: Expectation Compute $\ell_{t+1}(\theta) : \theta \mapsto \sum_{k=1}^{n} \mathbb{E}_{\nu_{k,\theta_t}}[\log q(\boldsymbol{x}_k, \boldsymbol{Z}; \theta)]$.
M: Maximisation Set $\theta_{t+1} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \ell_{t+1}(\theta)$.

The models we have discussed for ICA are exponential, in the sense that the joint distribution of the hidden variables for a given parameter can be put in form

$$\log q(\boldsymbol{x}, \boldsymbol{z}; \theta) = \phi(\theta) \cdot \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{z}) - \log C(\theta)$$

where $\boldsymbol{S}$ is a multidimensional sufficient statistic, ϕ is a fixed, vector-valued function of the parameters, C is a normalising constant and the dot refers to the usual Euclidean dot product. This implies that

$$\ell_{t+1}(\theta) = \phi(\theta) \cdot \Big(\sum_{k=1}^{n} \mathbb{E}_{\nu_{k,\theta_t}} \boldsymbol{S} \Big) - n \log C(\theta).$$

Thus, the E-step only requires to compute the conditional expectations of the sufficient statistic, and the M-step is equivalent to maximum likelihood for a fully observed model, with the empirical expectation of the sufficient statistic equal to

$$\bar{\boldsymbol{S}}_{\boldsymbol{t+1}} = \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}_{\nu_{k,\theta_t}} \boldsymbol{S}.$$

This is an important property (satisfied by our models) for the numerical feasibility of the EM.

However, this is not enough, since one also must be able to explicitly compute the conditional expectations. For several of our models, there is no close form expression for the densities $\nu_{k,\theta}$. For others, like IFA, for which such an expression can be derived, its computational complexity is exponential in the number of components which rapidly becomes intractable (details are given in the appendix).

A common way to overcome this difficulty is to approximate this conditional distributions by a Dirac measure at their mode. The resulting algorithm is sometimes called EM-MAP or FAM-EM (for "Fast Approximation with Mode") [8, 14]. At each iteration of the algorithm, one computes the most likely hidden variables $\hat{\boldsymbol{z}}_{\boldsymbol{k}}$ $\forall 1 \leq k \leq n$ with respect to the current parameters :

$$\hat{\boldsymbol{z}}_{t,k} = \underset{\boldsymbol{z}}{\operatorname{argmax}} \left[\log(q(\boldsymbol{z} | \boldsymbol{X} = \boldsymbol{x}_k, \theta_t)) \right] . \tag{3.3}$$

The M-step then maximises the likelihood for the "completed observations" $\boldsymbol{x}^{*n}$ and $\hat{\boldsymbol{z}}_{t,1}, \ldots, \hat{\boldsymbol{z}}_{t,n}$.

The statistical quality of this approximation is unclear, since it estimates a number of parameters that scales like the number of observations. Consistency of the obtained estimator when n goes to infinity cannot be proved in general. Some experimental evidence of asymptotic bias is demonstrated in Section 5 below.

In spite of these remarks, this approach (or approaches similar to it) is the most common choice for training probabilistic ICA models [6, 15–17]. In the under-determined problem ($p >> d$), this algorithm has also been implemented in [1].

Although the conditional distribution is not explicit, it is still possible (as we shall see later) to sample from it. The conditional expectation of the sufficient statistics ($\bar{\boldsymbol{S}}_{t+1}$) can therefore be approximated by Monte-Carlo simulation, as proposed in [18, 19] with the MCEM (Monte Carlo EM) algorithm. The resulting method, however, is heavily computational. Also, there is no guarantee that the errors resulting from the approximation to the E-step will cancel out to provide an estimator converging to a local maximum of the likelihood.

In this regard, a more interesting procedure, which has been proposed in [9], is a stochastic approximation of the EM algorithm, called SAEM. It replaces the E-step by a stochastic approximation step for the conditional likelihood (or, in practice, for the conditional expectation of the sufficient statistics), on which the M-step is based. More precisely, based on a sequence Δ_t of positive numbers decreasing to 0, the algorithm iterates the following two steps (assuming the tth iteration) :

SAE step For $k = 1, \ldots, n$, sample a new hidden variable $\boldsymbol{z}_{t+1,k}$ according to the conditional distribution ν_{k,θ_t} and define

$$\ell_{t+1}(\theta) = \ell_t(\theta) + \Delta_t \left(\sum_{k=1}^n \log q(\boldsymbol{x}_k, \boldsymbol{z}_{t+1,k}; \theta) - \ell_t(\theta) \right).$$

M step Set

$$\theta_{t+1} = \operatorname*{argmax}_{\theta \in \Theta} \ell_{t+1}(\theta).$$

For exponential families, the SAE step is more conveniently (and equivalently) replaced by an update of the estimation of the conditional expectation of the sufficient statistics, namely

$$\bar{\boldsymbol{S}}_{t+1} = \bar{\boldsymbol{S}}_t + \Delta_t \left(\frac{1}{n} \sum_{k=1}^n \boldsymbol{S}(\boldsymbol{x}_k, \boldsymbol{z}_{t+1,k}) - \bar{\boldsymbol{S}}_t \right),$$

with

$$\ell_{t+1}(\theta) = \phi(\theta) \cdot \bar{\boldsymbol{S}}_{t+1} - \log C(\theta)$$

being maximised in the M-step. Note that this algorithm is fundamentally distinct from the SEM method [20] in which the E step directly defines $\ell_{t+1}(\theta) = \sum_{k=1}^n \log q(\boldsymbol{x}_k, \boldsymbol{z}_{t+1,k}; \theta)$.

A final refinement may be required in the SAEM algorithm when directly sampling from the posterior distribution is unfeasible, or inefficient, but can be done using Markov Chain Monte Carlo (MCMC) methods. In this situation, there exists, for each θ and $\boldsymbol{x}$, a transition probability $z \mapsto \Pi_{\boldsymbol{x},\theta}(z, \cdot)$ such that the associated Markov chain is ergodic and has the posterior probability $q(\cdot|\boldsymbol{X} = \boldsymbol{x}; \theta)$ as stationary distribution. The corresponding variant of the SAEM (which we shall still call SAEM) replaces the direct sampling operation

$$\boldsymbol{z}_{t+1,k} \sim \nu_{k,\theta_t} = q(\cdot|\boldsymbol{X} = \boldsymbol{x}_k, \theta_t)$$

by a single Markov chain step

$$\boldsymbol{z}_{t+1,k} \sim \Pi_{\boldsymbol{x}_k,\theta_t}(\boldsymbol{z}_{t,k}, \cdot).$$

This procedure has been introduced and proved convergent for bounded missing data in [21]. This result has been generalised to unbounded hidden random variables in [10].

To ensure the convergence of this algorithm in the non-compact case (which is our case in the models above), one needs, in principle, to introduce a truncation on random boundaries as in [10]. This would yield add a step between the stochastic approximation and the maximisation steps, with the following *truncation step*. Let $\mathcal{S}$ be the range of the sufficient statistic, S. Let $(\mathcal{K}_q)_{q \geq 0}$ be an increasing sequence of compact subsets of $\mathcal{S}$ such as $\cup_{q \geq 0} \mathcal{K}_q = \mathcal{S}$ and $\mathcal{K}_q \subset \text{int}(\mathcal{K}_{q+1}), \forall q \geq 0$. Let $(\delta_t)_t$ a decreasing sequence of positive numbers. If $\bar{S}_{t+1}$ wanders out of $\mathcal{K}_{t+1}$ or if $|\bar{S}_{t+1} - \bar{S}_t| \geq \delta_t$ then the algorithm is reinitialised in a fixed compact set.

More details can be found in [22, 10]. In practice, however, our algorithms work properly without this technical hedge.

3.3. Application to our models

To complete the description of the SAEM algorithm for a given model, it remains to make explicit (i) the specific form of the sufficient statistic $\boldsymbol{S}$; (ii) the corresponding maximum likelihood estimate for complete observations; and (iii) the transition kernel for the MCMC simulation. Formulae for (i) and (ii) are provided in the Appendix for the ICA models we have considered here. For (iii), we have used a Metropolis-Hastings procedure, looping over the components, that we now describe (sometime called Metropolis-Hastings within Gibbs Sampling). This is for a fixed observation $\boldsymbol{x}_k$ and parameter θ, although we do not let them appear in the notation. So we let $\nu(.) = \nu_{k,\theta}$ be the probability that needs to be sampled from.

In the Metropolis-Hastings procedure, one must first specify a candidate transition probability $\rho(\boldsymbol{z}, \tilde{\boldsymbol{z}})$. A Markov chain $(\boldsymbol{Z}_t, t = 0, 1, \ldots)$ can then be defined by the two iteration steps, given $\boldsymbol{Z}_t$:

1. Sample $\boldsymbol{z}$ from $\rho(\boldsymbol{Z}_t, \cdot)$.
2. Compute the ratio
$$r(\boldsymbol{Z}_t, \boldsymbol{z}) = \frac{\nu(\boldsymbol{z})\rho(\boldsymbol{z}, \boldsymbol{Z}_t)}{\nu(\boldsymbol{Z}_t)\rho(\boldsymbol{Z}_t, \boldsymbol{z})}$$
and set $\boldsymbol{Z}_{t+1} = \boldsymbol{z}$ with probability $\min(1, r)$ and $\boldsymbol{Z}_{t+1} = \boldsymbol{Z}_t$ otherwise.

An interesting special case is when ρ corresponds to a Gibbs sampling procedure for the prior distribution, $q_m(\mathbf{z}; \theta)$. Given the current simulation $\boldsymbol{z}$, one randomly selects one component z^j and generate $\tilde{\boldsymbol{z}}$ by only changing z^j, replacing it by $\tilde{z}^j$ sampled from the conditional distribution $q_m(\tilde{z}^j | z^i, i \neq j; \theta)$. In this case, it is easy to see that the ratio r is then given by

$$r(\tilde{\boldsymbol{z}}, \boldsymbol{z}) = \frac{q(\boldsymbol{x_k}|\tilde{\boldsymbol{z}})}{q(\boldsymbol{x_k}|\boldsymbol{z})}.$$

The Markov kernel is then built by successively applying the previous kernel to each component.

Our implementation follows this procedure whenever the current set of parameters leads to an irreducible transition probability ρ. This is always true, excepted for the censored models, in which parameters $\alpha \in \{0, 1\}$ or $\gamma \in \{0, \frac{1}{2}\}$ are degenerate and must be replaced by some fixed values α_0 and γ_0 in the definition of ρ.

4. Reconstruction

Assuming that the parameters in the model are known or have been estimated, the reconstruction problem consists in estimating the hidden coefficients of the independent components, $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$, based on the observation of $\boldsymbol{x} \in \mathbb{R}^d$. (We briefly present the way to get these coefficients with the presented models, however, this is not our main concern in this paper.)

With probabilistic ICA, this is not as straightforward as with complete ICA, for which the operation only requires solving a linear system. A natural approach is maximum likelihood, i.e., (with our notation) find $\hat{\boldsymbol{z}} = argmax_{\boldsymbol{z}}\, \phi(\theta) \cdot S(\boldsymbol{x}, \boldsymbol{z})$ and deduce $\hat{\boldsymbol{\beta}}$ from it.

This maximisation is not explicit, although simpler for our two first models. Indeed, for Log-ICA, this requires to minimise

$$\frac{1}{2\sigma^2}|\boldsymbol{x} - \boldsymbol{A\beta}|^2 + 2\sum_{j=1}^{p} \log(e^{\beta^j} + e^{-\beta^j}).$$

(We take $\boldsymbol{\mu}_0 = 0$ in this section, replacing, if needed $\boldsymbol{x}$ by $\boldsymbol{x} - \boldsymbol{\mu}_0$.)

The Laplacian case, Lap-ICA, gives

$$\frac{1}{2\sigma^2}|\boldsymbol{x} - \boldsymbol{A\beta}|^2 + \sum_{j=1}^{p} |\beta^j|.$$

Both cases can be solved efficiently by convex programming. The Laplacian case is similar (up to the absence of normalisation of the columns of $\boldsymbol{A}$) to the Lasso regression algorithm [23], and can be minimised using an incremental procedure on the set of vanishing β^j's [24].

The EG-ICA problem requires to minimise

$$\frac{1}{2\sigma^2}|\boldsymbol{x} - \sum_{j=1}^{p} s^j y^j \boldsymbol{a}_j|^2 + \sum_{j=1}^{p} s^j + \frac{1}{2}\sum_{j=1}^{p}(y^j - \mu)^2\,,$$

with $s^1, \ldots, s^p \geq 0$. This is not convex, but one can use in this context an alternate minimisation procedure, minimising in $\boldsymbol{y}$ with fixed $\boldsymbol{s}$ and in $\boldsymbol{s}$ with fixed $\boldsymbol{y}$. The first problem is a straightforward least squares and the second requires quadratic programming.

The other models are more complex, because solving them all involve some form of quadratic integer programming, the general solution of which being NP-complete. When dealing with large

numbers of components, one must use generally sub-optimal optimisation strategies (including local searches) that have been developed for this context (see [25], for example).

The symmetrized IFA model leads to minimise

$$\frac{1}{2\sigma^2}|\boldsymbol{x}-\boldsymbol{A\beta}|^2-\sum_{j=1}^{p}\left(\frac{1}{2}(\beta^j-b^j m_{t^j})^2\right)+\sum_{j=1}^{p} logw_{t^j}\,.$$

with respect to $\boldsymbol{\beta}$, the unobserved configuration of labels $\boldsymbol{t}$, and the sign change $\boldsymbol{b}$. When labels and signs are given, the problem is quadratic in $\boldsymbol{\beta}$. For small dimensions, it is possible to make an exhaustive search of all $(2K+1)^p$ possible configurations of labels and signs.

For the BG-ICA, we must minimise

$$\frac{1}{2\sigma^2}|\boldsymbol{x}-\sum_{j=1}^{p} b^j y^j \boldsymbol{a}_j|^2+\rho\sum_{j=1}^{p} b^j+\frac{1}{2}\sum_{j=1}^{p}(y^j-\mu)^2\,,$$

with $\rho=\log((1-\alpha)/\alpha)$ and $b^j\in\{0,1\}$. The minimisation in $\boldsymbol{b}$ is a $(0,1)$-quadratic programming problem, an exhaustive search being feasible for small p. Given $\boldsymbol{b}$, the optimal $\boldsymbol{y}$ is provided by least squares.

Concerning the EBG-ICA, we must minimise

$$\frac{1}{2\sigma^2}|\boldsymbol{x}-\sum_{j=1}^{p} s^j b^j y^j \boldsymbol{a}_j|^2+\sum_{j=1}^{p} s^j+\rho\sum_{j=1}^{p} b^j+\frac{1}{2}\sum_{j=1}^{p}(y^j-\mu)^2.$$

with $\rho=\log((1-\alpha)/\alpha)$, $s^1,\ldots,s^p>0$ and $b^j\in\{0,1\}$. This is again a $(0,1)$-quadratic programming problem in $\boldsymbol{b}$ and, given $\boldsymbol{b}$, the optimal $\boldsymbol{y}$ and $\boldsymbol{s}$ are computed similarly to the EG-ICA model.

With ET-ICA, the objective function is

$$\frac{1}{2\sigma^2}|\boldsymbol{x}-\sum_{j=1}^{p} s^j y^j \boldsymbol{a}_j|^2+\sum_{j=1}^{p} s^j+\rho\sum_{j=1}^{p}|y^j|$$

with $\rho=\log((1-2\gamma)/2\gamma)$, $y^1,\ldots,y^p\in\{-1,0,1\}$ and $s^1,\ldots,s^p>0$. This is a quadratic integer programming in $\boldsymbol{y}$, with a complexity of 3^p for an exhaustive search. Given $\boldsymbol{b}$, computing $\boldsymbol{s}$ is a standard quadratic programming problem.

The TE-ICA problem, requiring to minimise

$$\frac{1}{2\sigma^2}|\boldsymbol{x}-s\sum_{j=1}^{p} y^j \boldsymbol{a}_j|^2+s+\rho\sum_{j=1}^{p}|y^j|\,,$$

is slightly simpler since, in this case, the computation of $s\geq 0$ given y is straightforward.

The TEoff-ICA model involves a third hidden variable μ. This leads to the following objective function to minimise both in s, $\boldsymbol{y}$ and μ:

$$\frac{1}{2\sigma^2}|\boldsymbol{x}-\boldsymbol{\mu}-s\sum_{j=1}^{p} y^j \boldsymbol{a}_j|^2+s+\rho\sum_{j=1}^{p}|y^j|,$$

with $\boldsymbol{\mu}=(\mu,\ldots,\mu)\in\mathbb{R}^d$, and $s>0$. Given $\boldsymbol{\mu}$ the minimisation with respect to s and $\boldsymbol{y}$ is done as in the previous TE-ICA model. The minimisation over μ has a closed form:

$$\mu=\frac{1}{d}\sum_{i=1}^{d}\left(\boldsymbol{x}^j-s\sum_{j=1}^{p} y^j \boldsymbol{a}_{i,j}\right)\,.$$

5. Experiments

5.1. Synthetic image data

We first provide an experimental analysis using synthetic data, which allows us to work in a controlled environment with a known ground truth. In this setting, we assume that the true distribution is the Bernoulli-Gaussian (BG) model, with two components ($p = 2$). The probability α of each component to be "on" is set to 0.8. We run experiments based on 30, 50 or 100 observations, and vary the standard deviation of the noise using $\sigma = 0.1,\ 0.5,\ 0.8,\ 1.5$.

The components are represented as two-dimensional binary images (grey levels being either 0 or 1). The first one is a black image (grey level equals 0) with a white cross (grey level equals 1) in the top left corner. The second one has a white square (same grey level) in the bottom right corner instead. These two images are shown in Figure 1. Figure 2 presents 30 images sampled from the model with different noise levels. The training sets were sampled once, and used in all the comparative experiments. We used a fixed colour map for all figures to allow for comparisons across experiments (this explains why the patterns in Figure 1 appear as grey instead of white).

FIG 1. *Two decomposition images used for synthetic data.*

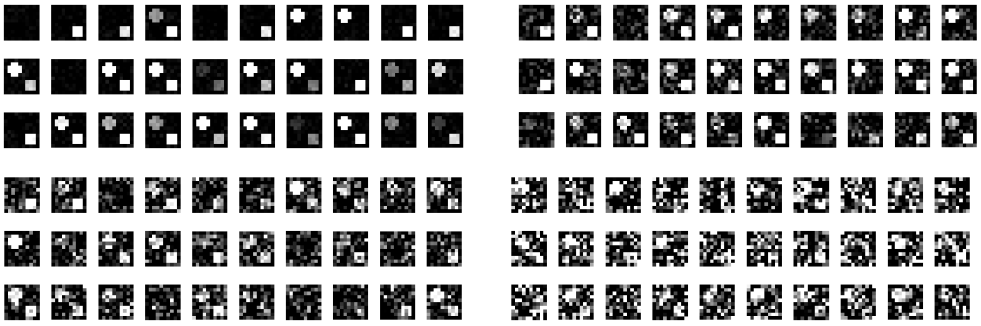

FIG 2. *Samples of the training sets used for synthetic data with different level of noise. From left to right and top to bottom:* $\sigma = 0.1,\ 0.5,\ 0.8,\ 1.5$

We have compared the following estimations strategies: (1) FAM-EM algorithm [15–17] (which maximises the likelihood with respect to parameters and hidden variables together), with the Log-ICA model (Logistic distribution); (2) SAEM with the same Log-ICA model; (3) SAEM for the same model, and (4) EM with the IFA model [12, 26]; (5) SAEM for the true BG-ICA model; (6) finally, we also ran a standard ICA decomposition (using fast-ICA [27]) with a requirement of computing only two components (with a preliminary dimension reduction based on PCA). (3) and (4) are theoretically equivalent in this framework, and our experiments simply check that it remains so experimentally. We strengthen the fact that the EM algorithm for the IFA model is only feasible for a reasonably small number of components, p, and number of mixtures, K (with a complexity in K^p), whereas this limitation does not apply to the SAEM algorithm (see Appendix for more details). For other alternative approaches to the EM for the IFA model (including the use of the FAM-EM strategy), see [28, 7, 29, 4, 5]. The fast-ICA algorithm used in (6) is non-parametric (and maximises an approximation of the so-called negentropy of the model).

Our first remark is that the SAEM algorithm has demonstrated excellent stability and convergence properties in these experiments. The estimated components are fairly consistent with the ground truth, even when the model used for the estimation differs from the true one (note that the decomposition matrix is estimated up to a permutation of its columns). This does not apply to

the "FAM-EM" algorithm (which maximises the likelihood with respect to parameters and hidden variables together), which significantly degenerates in the presence of high noise. Figures 3 and 4 present the results of these experiments. The coupling of models and algorithm are presented in rows and the columns correspond to increasing noise level.

The estimation of the components is quite accurate for all models and algorithms with low noise levels. FAM-EM clearly breaks down when this level increases, and adding more sample in the training set does not seem to help. On the other hand, the SAEM algorithm (and the EM applied to the IFA model) reaches decomposition vectors that are consistent with the ground truth. Increasing the number of images in the training set improves the estimation, as could be expected with maximum likelihood estimators. The EM and SAEM algorithms for the IFA model provide similar results. Fast-ICA also breaks down in the presence of high noise. This can be due to the inaccuracy of PCA dimension reduction. We also experienced numerical failures when running the publicly available software in such extreme situations (we had, in fact, to resample a new 100-image training set to be able to present results from this method).

We also evaluated the accuracy of the estimation of σ^2. The results are presented in Table 1. A surprising result is that σ^2 is always well estimated even when the decomposition vectors are not. This is an important observation which indicates that one should not evaluate the final convergence of any algorithm based on the convergence of σ^2 only.

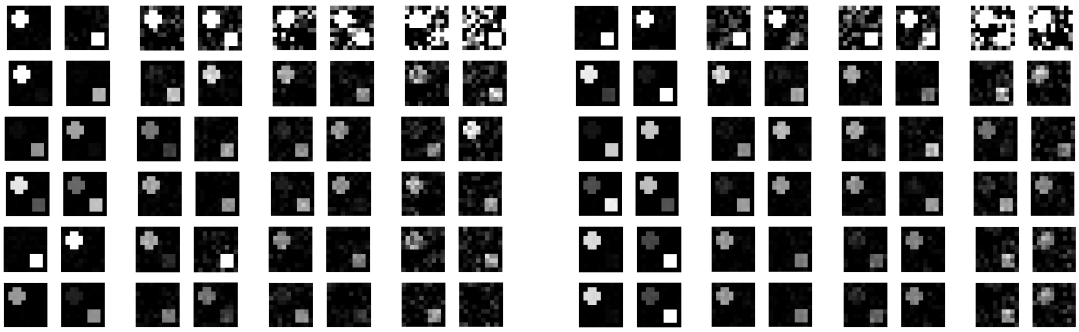

FIG 3. *Estimated decomposition images with different models and algorithm: 1st row: Fast Approximation with Mode (FAM-) EM algorithm with the Logistic distribution. 2nd row: SAEM algorithm with the Logistic distribution. 3rd row: SAEM algorithm with the IFA model. 4th row: EM algorithm with the IFA model. 5th row: SAEM algorithm with the BG-ICA model. 6th row: FastICA two most important decomposition vectors. The experiments are done with different noise levels:* $\sigma = 0.1,\ 0.5,\ 0.8,\ 1.5$ *(column 1 to 4 respectively) and* 30 *images in the training set (left) and (column 5 to 8 respectively) and* 50 *images in the training set (right).*

Model + algorithm	True σ^2	Log + FAM-EM	Log + SAEM	IFA + EM	IFA + SAEM	BG + SAEM
30 images in the training set	0.001	0.0088	0.0086	0.0097	0.0089	0.0087
	0.2500	0.2253	0.2224	0.2240	0.2410	0.2226
	0.6400	0.5685	0.5577	0.5534	0.6092	0.5569
	2.2500	2.0375	1.9978	2.1199	2.0735	2.0009
Model + algorithm	True σ^2	Log + FAM-EM	Log + SAEM	IFA + EM	IFA + SAEM	BG + SAEM
50 images in the training set	0.001	0.0095	0.0092	0.0095	0.0094	0.0092
	0.2500	0.2400	0.2399	0.2363	0.2524	0.2399
	0.6400	0.5831	0.5798	0.6381	0.6429	0.5795
	2.2500	2.1544	2.1377	2.2061	2.2112	2.1366
Model + algorithm	True σ^2	Log + FAM-EM	Log + SAEM	IFA + EM	IFA + SAEM	BG + SAEM
100 images in the training set	0.001	0.0176	0.0097 0.0095	0.0098	0.0097	
	0.2500	0.2432	0.2459	0.2455	0.2564	0.2456
	0.6400	0.6225	0.6282	0.6336	0.6388	0.6280
	2.2500	2.1268	2.1479	2.1767	2.1970	2.1490

TABLE 1

Estimated noise variance with the different models and the two different algorithms for 30, 50 *and* 100 *images in the training set. These variances correspond to the estimated decomposition vectors presented in Figures 3, and 4*

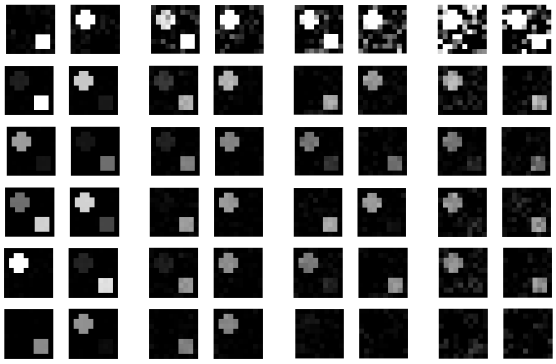

FIG 4. *Estimated decomposition images with different models and algorithm: 1st row: FAM-EM algorithm with the Logistic distribution. 2nd row: SAEM algorithm with the Logistic distribution. 3rd row: SAEM algorithm with the IFA model. 4th row: EM algorithm with the IFA model. 5th row: SAEM algorithm with the BG-ICA model. 6th row: FastICA two most important decomposition vectors. The experiments are done with different noise levels:* $\sigma = 0.1,\ 0.5,\ 0.8,\ 1.5$ *(column 1 to 4 respectively) and* 100 *images in the training set.*

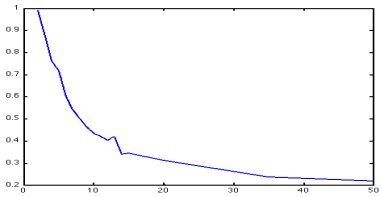

FIG 5. *Estimated component activation probability* (α) *as a function of the model size for a Bernoulli Gaussian model. Ground truth is* $p = 8$ *and* $\alpha = 0.5$.

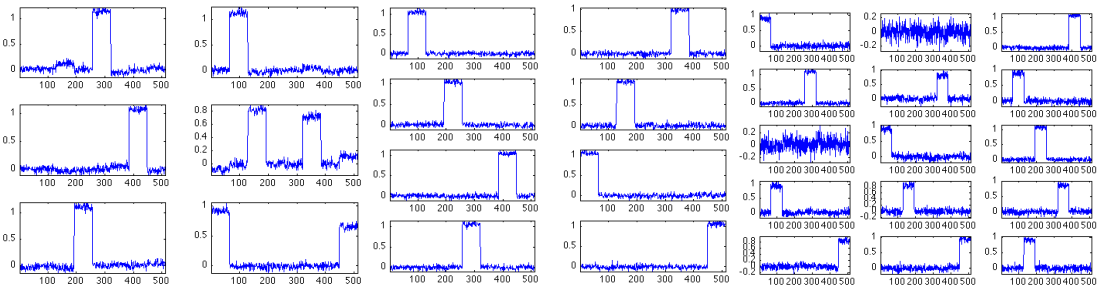

FIG 6. *Estimated components with probabilistic ICA. The ground-truth model has eight components. From left to right: components estimated with* $p = 6$, *8 and 15.*

5.2. Effect of the number of estimated components

We also illustrate how the estimation of the censoring coefficient evolves with the number of components. In this experiment, we have generated 1000 sample of a shifted Bernoulli-Gaussian model (see section 2.9), with 8 components (the components being represented as indicators of 8 non-overlapping intervals). The true value of α is 0.5, and we took $\mu = 2$. In figure 5, we plot the value of the estimated α as a function of the number of components in the model, p. We can see that this value seems to decrease to zero, at a rate which is however not linear in $1/p$. The expected number of non-zero components grows from 2 for $p = 2$, to 4 when $p = 8$ (correct value), to about 10 when $p = 50$. The estimated components for $p = 6$, 8 and 15 are plotted in Figure 6. This illustrates the effect of under-dimensioning the model, in which some of the estimated components must share some of the features of several true components, and of over-dimensioning, in which some of the estimates components are essentially noise (clearly indicating over-fitting of the data), while some other estimated components, which correspond to true ones, are essentially repeated. Components are correctly estimated when the estimated model coincides with the true model ($p = 8$).

Although we are not addressing the estimation of the number of components in this paper, these results clearly indicate that this issue is important. From a computational point of view, it would not be difficult to complete the model with a prior distribution on the parameters and on the model size, and adapt the SAEM algorithm accordingly.

5.3. Handwritten digits

We now test our algorithms on some 2D images. The first training set we use is the USPS database, which contains 7291 grey-level images of size 16×16. We used the whole database as training set and computed 20 decomposition vectors. Some images from this data set are presented in Figure 7 (left).

The different decomposition vectors and the estimated means (when it is a parameter) are presented in Figure 8. Each set of 20 images (10 times 2 lines) on the right column shows one run of one algorithm corresponding to selected models from the previous ones. We have selected the most representative results, the other ones were similar to one of the one shown.

The results are interesting. They demonstrate, in particular, the advantage of modelling component coefficients that can vanish with positive probability (BG and ET-ICA). With these models, many decomposition vectors represent well-formed digits, whereas the decomposition vectors for other models mix several digits more often. The fact that the model can cancel some independent component allows these very typical decomposition vectors to appear. When considering a data set such as USPS where one image in one class is not easily expressed as a mixture of images from other classes, these binary or ternary models seem to be adequate.

Note that the USPS data set does not have the same amount of images of each digit. There are about twice as many 0s or 1s as other digits. This fact explains the "bias" one can see on the mean, on which the shape of the zero is noticeable. In all experiments the trace of each digit can be (more or less easily) detected in at least one of the components, at the exception of digit 2. This is probably due to the large geometrical variability of the 2s, which is much higher than other digits (changes of topology -loop or not, changes in global shape) and therefore difficult to capture.

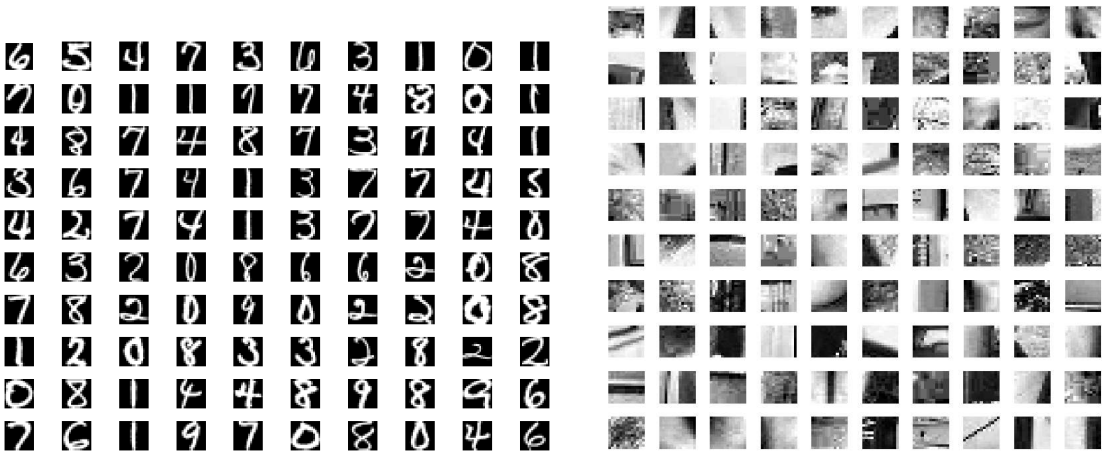

FIG 7. *100 images randomly extracted from the USPS database (left) and from the face category in the Caltech101 data set (right).*

5.4. Face images

We have run a similar experiment on a data set of face images (taken from the Caltech101 dataset). Each of these images has been decomposed into patches of size 13×13 some of them are presented in Figure 7 (right). The resulting database contains 499,697 small images and we estimated 20 decomposition vectors. Results are presented in Figure 9. The patterns which emerge from the estimations are quite similar from one model to another: vertical, horizontal and diagonal separation of the image into black and white, blobs, regular texture like a regular mesh, etc.

We also ran the same estimation with two of the previous models looking for 100 decomposition vectors. The results are presented in Figure 10. We selected the Log and BG-ICA since one has a

FIG 8. *Results of the independent component estimation on the USPS database using four selected models. The training set is composed of* 7291 *images containing the 10 digits randomly spread. Left column: mean image* μ_0. *Right column: 20 estimated decomposition vectors.*

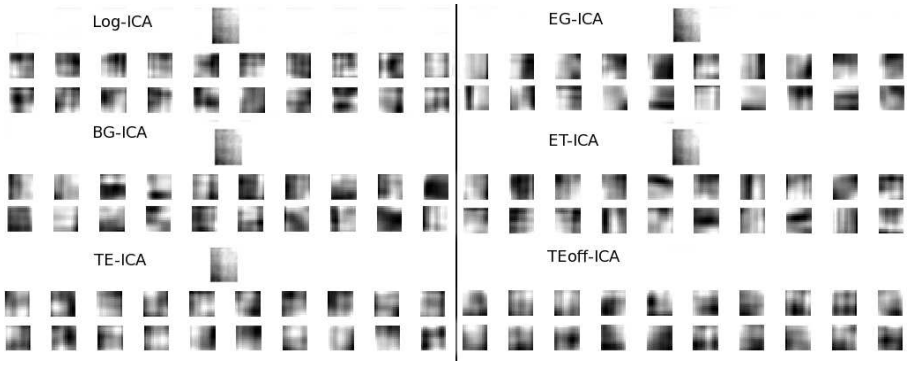

FIG 9. *Decomposition vectors from six selected models. From left to right and top to bottom: Log-ICA, Lap-ICA, EG-ICA, BG-ICA, EBG-ICA, ET-ICA, TE-ICA, TEoff-ICA. For each model the top row is the mean image and the bottom rows are the 20 corresponding decomposition vectors.*

continuous density and the second has a discrete one. The results are rather different. While the Log-ICA model tends to catch some textures, the BG-ICA captures some shapes. In this example, as well as with the digit case, the sparsity of the decomposition makes sense and plays an important role. This database is composed of discrete features which can hardly be approximated by a linear combination of continuous patterns. The models generating sparse representations again seems to be better adapted to this kind of data.

5.5. Anatomical surfaces

We finally consider a data set containing to a family of 101 hippocampus surfaces that have been registered to a fixed template using Large Deformation Diffeomorphic Metric Mapping [30–33]. We here analyse the logarithm of the Jacobian determinant of the estimated deformations, represented (for each image) as a scalar field over the surface of the template, described by a triangulated mesh. These vectors have fixed length ($d = 3223$), equal to the number of vertices in the triangulation.

The 101 subjects in the dataset are separated in 3 groups with 57, 32 and 12 patients, containing healthy patients in the first group and patients with Alzheimer's disease and semantic dementia (denoted the AD group later) at different stages in the last two groups.

Using our algorithm, we have computed $p = 5$ decomposition vectors based on the complete data set. Figures 12 to 14 present these decomposition vectors mapped on the meshed hippocampus for six selected models. The estimated mean is shown on the left side and the five corresponding

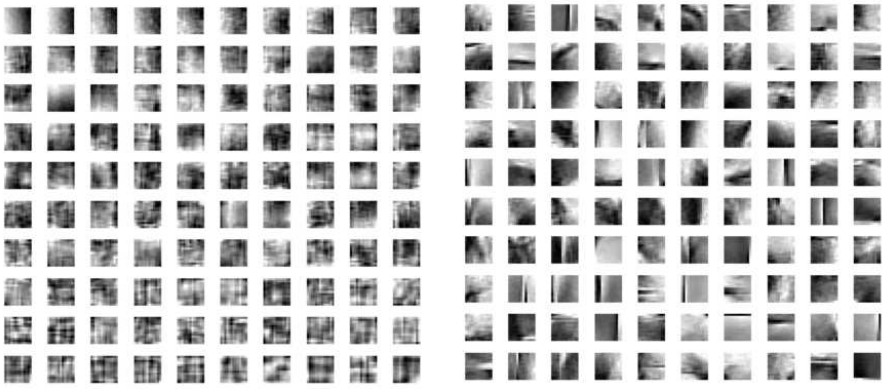

FIG 10. 100 *decomposition vectors from 2 models. Left: Log-ICA. Right: BG-ICA.*

decomposition vectors are on the right side. Images are presented with different colour maps, to facilitate the visualisation of the patterns. In particular, even if the means seem to contain a lot of information, they vary on a very small scale compared to all the decomposition vectors (they are actually close to 0).

Although results vary with the chosen model, we can see common features emerging. First of all, the means are very close to each other. The patterns which we can notice on each of them is the same. For example, there is a noticeable contraction on the top part and an extension on the bottom left side of the shape. These deformations however have a small amplitude and can be interpreted as the "bias" of the training set. Concerning the decomposition vector themselves, the pattern of the first vector of the Logistic model is present in all other models (for example in position 1 for the Laplacian, EG, TE and TEoff models (not shown here), 4 for the BG model, 5 for the EBG (not shown here) and 2 for the ET model). Other patterns occurs also like a contraction or a growth of the tail part (in vector 3 of Log, Lap, EG, BG, EBG, TEoff (not shown here) and 5 of TE) or on the bottom of the left part of the image (in vectors 4 and 5 of Log, 5 of Lap, EG, BG and TEoff (not shown here) and in vector 1 otherwise). These common features seem to be characteristic of this population.

In tables 2 and 3, we provide the p-value obtained from the comparison of the five ICA coefficients (β) among the three subgroups. The test is based on a Hoteling T-statistic evaluated on the coefficients, the p-value being derived using permutation sampling. The algorithm we propose in this paper is stochastic and is supposed to converge toward a critical point of the likelihood of the observations. However, we do not control which critical point we reach and in addition, because of the stochastic character of the procedure, different runs of the algorithm starting from the same initial point can lead to different limits. To control the effect of this variability, we ran the algorithm for each model 50 times, with the same initial conditions, and compute an average and a standard deviation of the p-values.

The test is performed for two different comparisons: first we compare the healthy group with respect to the two pathological groups. This is what is shown in Table 2. The second test compares the healthy group with the group of 32 mild AD patients. The results are presented in Table 3.

The results are mostly significant. Indeed, most of all methods yield p-values under 1% when we compare the control population to the AD groups and less than 3% for the comparison of the control versus mild AD.

The only model which does not yield significant p-values is the offset case. Both the mean and standard deviation are high (even

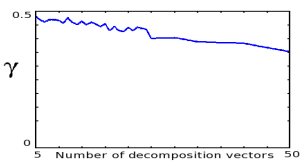

FIG 11. *Evolution of the probability of one component to activate or inhibit the corresponding decomposition vector in the ET-model with respect to the number of decomposition vectors. The training set is the set of* 101 *hippocampi.*

higher when we focus on the mild AD population). This suggests that this model on this database is unstable. One run can lead to significant decomposition vectors and a second one can lead to very different results. This particular model does not seem to be adapted to this particular type of data contrary to the USPS database for example. The mean is very close to zero and is therefore not a relevant variable for this application. The additional variability in the model may have an adverse effect on the estimation.

Figure 11, provides some insight in the way components are turned on/off by the ET-ICA model, by plotting the estimated probability, $\gamma = P(Y_k^j = -1) = P(Y_k^j = 1)$, against the number of decomposition vectors, p. As already noticed in section 5.2, for small p, all components are needed, yielding $\gamma \simeq 1/2$. When more components are added, they do not need to appear all the time, yielding a decreasing value of γ.

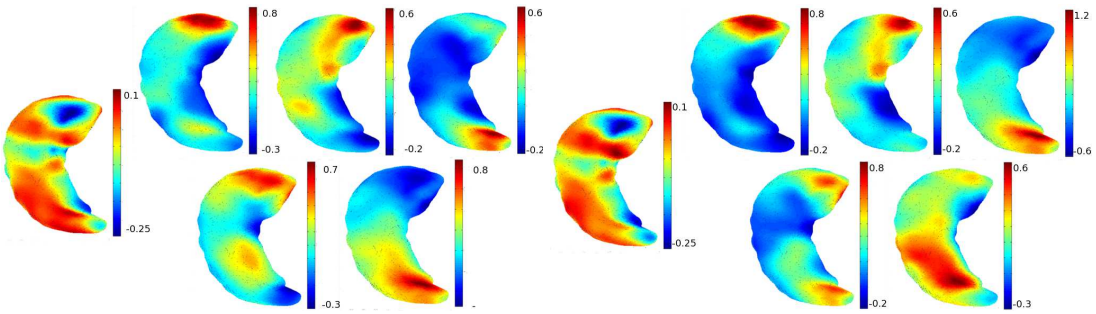

FIG 12. *Left: Mean (left) and 5 decomposition vectors estimated with the Log-ICA model. Right: Mean (left) and 5 decomposition vectors estimated with the Lap-ICA model. Each image has its own colour map to highlight the major patterns.*

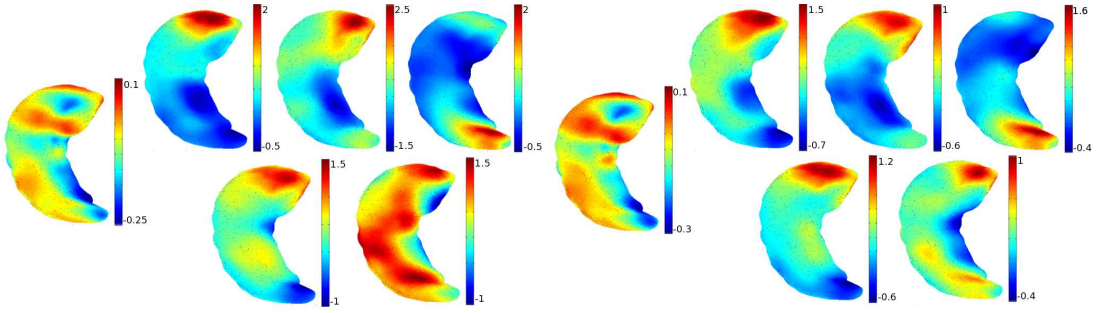

FIG 13. *Left: Mean (left) and 5 decomposition vectors estimated with the EG-ICA model. Right: Mean (left) and 5 decomposition vectors estimated with the BG-ICA model. Each image has its own colour map to highlight the major patterns.*

6. Conclusion and discussion

This paper presents a new solution for probabilistic independent component analysis. Probabilistic ICA enables to estimate a small number of features (compared to the dimension of the data) which characterise a data set. Compared to plain ICA, this reduction of dimension avoids the instability of the computation of the decomposition matrix when the number of observations is much smaller than their dimension (typical case in medical imaging). We have demonstrated that the Stochastic Approximation EM algorithm is an efficient and powerful tool which provides a convergent method that estimates the decomposition matrix. We have shown that this procedure does not restrict the large choice of distributions for the independent components, as illustrated by eight models with

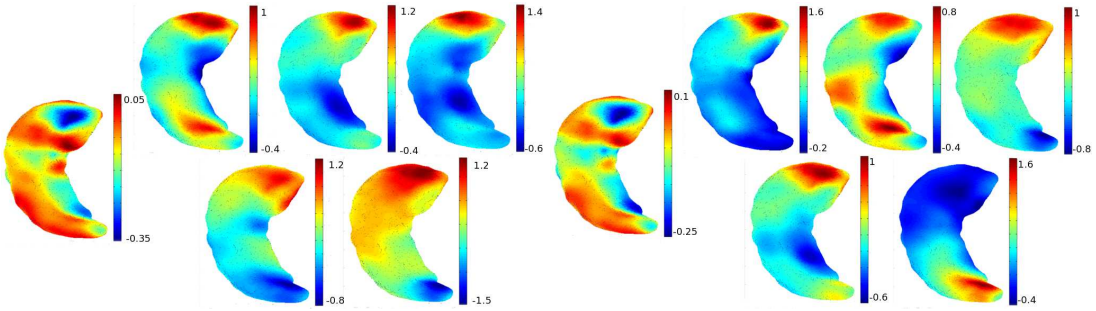

FIG 14. *Left: Mean (left) and 5 decomposition vectors estimated with the ET-ICA model. Right: Mean (left) and 5 decomposition vectors estimated with the TE-ICA model. Each image has its own colour map to highlight the major patterns.*

Model	Log-ICA	Lap-ICA	EG-ICA	BG-ICA	EBG-ICA
Mean on log	0.31×10^{-3}	0.29×10^{-3}	0.27×10^{-3}	0.33×10^{-3}	0.9×10^{-3}
Std deviation on log	0.16×10^{-3}	0.19×10^{-3}	0.12×10^{-3}	0.25×10^{-3}	1.2×10^{-3}

Model	ET-ICA	TE-ICA	TEoff-ICA
Mean on log	0.27×10^{-3}	2.4×10^{-3}	7.57×10^{-2}
Std deviation on log	0.14×10^{-3}	2.9×10^{-3}	12.62×10^{-2}

TABLE 2

Mean and standard deviation of the p-values for the eight (plus IFA for comparison) models with the five decomposition vectors shown in Figures (12) to (14). The mean and the standard deviation are computed over 50 samples of the posterior distributions of the hidden variables to separate the first group (Control) with respect to the two others (AZ).

different properties, mixing continuous and discrete probability measures, that we have introduced and studied.

Future works will be devoted to the analysis of non-linear generative models that allow for the analysis of data on Riemannian manifolds, including the important case of shape spaces in which the models generate nonlinear deformation of given templates. Generalisations of the methods proposed in [8, 34] will be developed, in order to estimate both the templates and the generative parameters.

Appendix A: Proof of the sub-exponential tail of the EG-distribution

Let (Y, S) be a couple of independent random variable where Y and S have a standard normal distribution and an exponential distribution respectively. Let $\beta = Y\ S$ and assume $t > 0$ so that $\beta > t$ implies $Y > 0$. We have (letting $C = (2\pi)^{-1/2}$)

$$\begin{aligned}\mathbb{P}(\beta > t) &= \mathbb{P}(s > t/y, y > 0) = C \int_0^\infty \mathbb{P}\left(s > \frac{t}{y}\right) \exp\Big(-\frac{1}{2}y^2\Big) dy \\ &= C \int_0^\infty \exp\left(-\frac{1}{2}y^2 - \frac{t}{y}\right) dy\,.\end{aligned}$$

Let $h_t(y) = -\frac{1}{2}y^2 - \frac{t}{y}$. We will use the Laplace method to compute an equivalent of the previous integral. The function h_t is maximum for $y^* = t^{1/3}$ and its value is $h_t(y^*) = -\frac{3}{2}t^{2/3}$. Therefore,

$$\mathbb{P}(\beta > t) \sim C \exp\left(-\frac{3}{2}t^{2/3}\right) \int_0^\infty \exp\left(-\frac{3}{2}(y - t^{1/3})^2\right) dy\,.$$

Thanks to a change of variable, it is easy to show that :

$$\int_0^\infty \exp\left(-\frac{3}{2}(y - t^{1/3})^2\right) dy \sim -\frac{1}{3t^{1/3}} \exp\left(-\frac{3}{2}t^{2/3}\right)\,.$$

Model	Log-ICA	Lap-ICA	EG-ICA	BG-ICA	EBG-ICA
Mean on log	$9.0\ \times 10^{-3}$	$9.6\ \times 10^{-3}$	$8.3\ \times 10^{-3}$	$1.09\ \times 10^{-2}$	$1.87\ \times 10^{-2}$
Std deviation on log	$3.8\ \times 10^{-3}$	$4.8\ \times 10^{-3}$	$2.7\ \times 10^{-3}$	$7.6\ \times 10^{-3}$	$1.77\ \times 10^{-2}$

Model	ET-ICA	TE-ICA	TEoff-ICA
Mean on log	$8.9\ \times 10^{-3}$	$3.08\ \times 10^{-2}$	$14.87\ \times 10^{-2}$
Std deviation on log	$4.6\ \times 10^{-3}$	$2.88\ \times 10^{-2}$	$16.04\ \times 10^{-2}$

TABLE 3

Mean and standard deviation of the p-values for the eight models with the five decomposition vectors shown in Figures (12) to (14). The mean and the standard deviation are computed over 50 samples of the posterior distributions of the hidden variables to separate the first group (Control) with respect to the second one (mild AZ).

This yields

$$\mathbb{P}(\beta > t) = O\left(t^{-1/3}\exp\left(-\frac{3}{2}t^{2/3}\right)\right).$$

Note that the density of β, which is $g(\beta) = \int_0^\infty \exp\left(-\frac{1}{2}y^2 - \frac{\beta}{y}\right)\frac{dy}{y}$ has a singularity at $\beta = 0$.

Appendix B: Maximum Likelihood for the complete models

The M-step in our models requires solving the equation $E_\theta(\boldsymbol{S}) = [\boldsymbol{S}]$ where $[\boldsymbol{S}]$ is a prescribed value of the sufficient statistic (an empirical average for complete observations, or what we have denoted $\bar{\boldsymbol{S}}_{\boldsymbol{t}}$ in the M-step of the learning algorithm). In the next sections, we provide the expressions of $\boldsymbol{S}$ for the family of models we consider and give the corresponding solution of the maximum likelihood equations. Notice that these are closed form expressions, ensuring the simplicity of each iteration of the SAEM algorithm.

B.1. Log-ICA and Lap-ICA models

For these models, the log-likelihood is

$$-\sum_{j=1}^{p}\xi(\beta^j) - \frac{1}{2\sigma^2}|X - \boldsymbol{\mu}_0 - \sum_{j=1}^{p}\beta^j\boldsymbol{a}_j|^2 - \log C(\sigma^2, \boldsymbol{A})$$

where $\xi(\beta) = 2\log(e^\beta + e^{-\beta})$ in the logistic case, and $\xi(\beta) = |\beta|$ in the Laplacian case. As customary, and to lighten the formulae, we let $\beta^0 = 1$ and $\boldsymbol{a}_0 = \boldsymbol{\mu}_0$, so that $\boldsymbol{\beta}$ and $\boldsymbol{A}$ have size $d+1$, and remove $\boldsymbol{\mu}_0$ from the expressions for this model and the following ones. We will also leave to the reader the easy modifications of the algorithms in the case of shifted models described in section 2.9.

The likelihood can be put in exponential form using the sufficient statistic $\boldsymbol{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \boldsymbol{X}\boldsymbol{\beta}^T)$, from which the maximum likelihood estimator can be deduced using:

$$\begin{cases} \boldsymbol{A} &= [\boldsymbol{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}, \\ \sigma^2 &= \frac{1}{d}\left([|\boldsymbol{X}|^2] - 2\langle\boldsymbol{A}, [\boldsymbol{X}\boldsymbol{\beta}^T]\rangle_F + \langle\boldsymbol{A}^T\boldsymbol{A}, [\boldsymbol{\beta}\boldsymbol{\beta}^T]\rangle_F = [|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{\beta}|^2]/d\right). \end{cases}$$

where $\langle\cdot,\cdot\rangle_F$ refers to the Frobenius dot product between matrices (the sum of products of coefficients).

B.2. IFA model

The complete log $-$likelihood of the Independent Factor Analysis model for a single observation X is:

$$-\frac{1}{2\sigma^2}|X - \sum_{j=1}^{p}\beta^j\boldsymbol{a}_j|^2 - \frac{1}{2}\sum_{j=1}^{p}(\beta^j - b^j m_{t^j})^2 + \sum_{j=1}^{p}\log w_{t^j} - \log C(\boldsymbol{A}, \sigma, \boldsymbol{m}, \boldsymbol{w}).$$

This formulation leads to the following sufficient statistics:

$$S = \left(S_0 = \sum_{j=1}^{p} \mathbb{1}_{t_j=k}, S_1 = \sum_{j=1}^{p} \mathbb{1}_{t_j=k} b^j \beta^j, \boldsymbol{\beta}\boldsymbol{\beta}^T, \boldsymbol{X}\boldsymbol{\beta}^T \right) .$$

The estimator associated to averaged values of these statistics (denoted as above with brackets) is:

$$\left\{ \begin{array}{lcl} \boldsymbol{A} & = & [\boldsymbol{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}\,, \\ \sigma^2 & = & [|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{\beta}|^2]/d \\ m_k & = & [S_1]/[S_0]\,, \\ w_k & = & [S_0]/p\,. \end{array} \right.$$

For this model, it is also possible to compute the conditional distribution of the hidden variables, $\boldsymbol{\beta}, \boldsymbol{t}$ and $\boldsymbol{b}$ given observed values of X [12]. Indeed, for given $\boldsymbol{b}$ and $\boldsymbol{t}$, let $\mu_{\boldsymbol{b},\boldsymbol{t}} = (b^1 m_{t^1}, \ldots, b^p m_{t^p})$. Let $\Lambda = (\mathrm{Id}_{\mathbb{R}^p} + \frac{A^T A}{\sigma^2})$ and, for a given X $\mu_{\boldsymbol{b},\boldsymbol{t},X} = \Lambda(A^T X + \mu_{\boldsymbol{b},\boldsymbol{t}})$. Then, a rewriting of the likelihood above shows that the conditional distribution of $\boldsymbol{\beta}$ given $X, \boldsymbol{T}$ and $\boldsymbol{b}$ is Gaussian with mean $\mu_{\boldsymbol{b},\boldsymbol{t},X}$ and covariance Λ, and that the conditional distribution of $(\boldsymbol{t}, \boldsymbol{b})$ is the discrete distribution with

$$\pi(\boldsymbol{t}, \boldsymbol{b}|X) \propto \exp\Big(-\frac{1}{2}(|\mu_{\boldsymbol{b},\boldsymbol{t}}|^2 - (A^T X + \mu_{\boldsymbol{b},\boldsymbol{t}})^T \Lambda (A^T X + \mu_{\boldsymbol{b},\boldsymbol{t}}))\Big) \prod_{j=1}^{p} w_{t^j}\,.$$

Using these expressions, the E-step of the EM-algorithm can be computed exactly, but it requires computing all $(2K+1)^p$ conditional probabilities $\pi(\boldsymbol{t}, \boldsymbol{b}|X)$, which becomes intractable for large dimensions. In contrast, each step of the SAEM algorithm only requires sampling from the conditional distributions, and has complexity of order $p(2K+1)$.

The same remark on the feasibility of the EM algorithm holds for for all our models with discrete variables (BG-ICA, ET-ICA, etc.), for which the E-step of the algorithm can be made explicit by conditioning on the discrete variables, with a cost that grows exponentially in the number of components, whereas the sampling part of SAEM only grows linearly.

B.3. EG-ICA model

The likelihood is

$$-\frac{1}{2}\sum_{j=1}^{p}(Y^j)^2 - \frac{1}{2}\sum_{j=1}^{p} s^j - \frac{1}{2\sigma^2}|X - \sum_{j=1}^{p} s^j Y^j \boldsymbol{a}_j|^2 - \log C(\sigma^2, \boldsymbol{A})$$

with sufficient statistic $\boldsymbol{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \boldsymbol{X}\boldsymbol{\beta}^T)$ with $\beta^j = s^j Y^j$. The maximum likelihood then is

$$\left\{ \begin{array}{lcl} \boldsymbol{A} & = & [\boldsymbol{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1} \\ \sigma^2 & = & [|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{\beta}|^2]/d. \end{array} \right.$$

B.4. BG-ICA and EBG-ICA models

These two models have the same parameters and therefore the same function to maximise. The likelihood is

$$-\frac{1}{2}\sum_{j=1}^{p}(Y^j)^2 + \log\left(\frac{\alpha}{1-\alpha}\right)\sum_{j=1}^{p} b^j - \frac{1}{2\sigma^2}|X - \sum_{j=1}^{p} b^j Y^j \boldsymbol{a}_j|^2 - \log C(\sigma^2, \boldsymbol{A}, \mu, \alpha)$$

with sufficient statistic $\boldsymbol{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \boldsymbol{X}\boldsymbol{\beta}^T, \nu)$ with $\beta^j = b^j Y^j$ and $\nu = b^1 + \cdot + b^p$. The optimal parameters are

$$\left\{ \begin{array}{lcl} \boldsymbol{A} & = & [\boldsymbol{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}\,, \\ \sigma^2 & = & [|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{\beta}|^2]/d\,, \\ \alpha & = & [\nu]/p\,. \end{array} \right.$$

B.5. ET-ICA, TE-ICA and TEoff-ICA models

We turn to the ternary models which share the same parameters (up to μ_0 for the offset model). The likelihood to maximise is

$$\log\left(\frac{\gamma}{1-\gamma}\right)\sum_{j=1}^{d}|Y^j| - \frac{1}{2\sigma^2}|X - \sum_{j=1}^{p} s^j Y^j \boldsymbol{a}_j|^2 - \log C(\sigma^2, \boldsymbol{A}, \gamma)$$

with sufficient statistic $\boldsymbol{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \boldsymbol{X}\boldsymbol{\beta}^T, \zeta)$, $\beta^j = s^j Y^j$, $\zeta = |Y^1| + \cdots + |Y^p|$. The optimal parameters are

$$\left\{\begin{array}{lcl} \boldsymbol{A} & = & [\boldsymbol{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1} \\ \sigma^2 & = & [|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{\beta}|^2]/d\,. \\ \gamma & = & [\zeta]/p \end{array}\right.$$

The maximum likelihood estimator for the single scale model is given by the same formulae, using $\beta^j = sY^j$.

References

[1] O. Bremond, E. Moulines, J.-F. Cardoso, Séparation et déconvolution aveugle de signaux bruités: modélisatin par mélange de gaussiennes, GRETSI.

[2] M. Uzumcu, A. F. Frangi, J. H. Reiber, B. P. Lelieveldt, Independent component analysis in statistical shape models, SPIE Medical Image Analysis.

[3] A. Bell, T. Sejnowski, An information maximisation approach to blind separation and blind deconvolution, Neural Computation 7, 6, (1995) 1129–1159.

[4] E. Côme, Z. Cherfi, L. Oukhellou, P. Aknin, Semi-supervised ifa with prior knowledge on the mixing process. an application to railway device diagnosis, Proc of the International Conference on Machine Learning and Application.

[5] G. Varoquaux, S. Sadaghini, J. Poline, B. Thirion, A group model for stable multi-subject ica on fmri datasets, In Press, NeuroImage.

[6] A. Hyvarinen, Survey on independent component analysis, Neural Computing Surveys 2 (1999) 94–128.

[7] H. Valpola Lappalainen, P. Pajunen, Fast algorithms for bayesian independent component analysis, In Proc. of the Second International Workshop on Independent Component Analysis and Blind Signal Separation, ICA.

[8] S. Allassonnière, Y. Amit, A. Trouvé, Toward a coherent statistical framework for dense deformable template estimation, JRSS 69 (2007) 3–29.

[9] B. Delyon, M. Lavielle, E. Moulines, Convergence of a stochastic approximation version of the EM algorithm, Ann. Statist. 27 (1) (1999) 94–128.

[10] S. Allassonnière, E. Kuhn, A. Trouvé, Bayesian deformable models bulding via stochastic approximation algorithm: A convergence study, In Press in Bernoulli J.

[11] A. Kagan, Y. Linnik, C. Rao, Characterization problems in mathematical statistics, Wiley.

[12] H. Attias, Independent factor analysis, Neural Computation 11 (1999) 803–851.

[13] J. W. Miskin, D. MacKay, Ensemble learning for blind source separation, Independent Component Analysis: Principle and Practice S. Roberts and R. Everson (Eds), Cambridge University Press (2001) 209–233.

[14] S. Allassonnière, E. Kuhn, A. Trouvé, Map estimation of statistical deformable templates via nonlinear mixed effects models : Deterministic and stochastic approaches, in: X. Pennec, S. Joshi (Eds.), Proc. of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA), 2008.

[15] D. B. Grimes, R. P. Rao, Bilinear sparse coding for invariant vision, Neural Computation 17 (2005) 47–73.

[16] B. A. Olshausen, D. J. Field.

[17] Separating style and content with bilinear models, Neural Computation 12 Issue 6.

[18] M. A. Tanner, Tools for statistical inference, Springer-Verlag New York.
[19] G. C. Wei, M. A. Tanner, A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms, Journal of the American Statistical Association 85, No. 411 (1990) 699–704.
[20] G. Celeux, J. Diebolt, The sem algorithm : a probabilistic teacher algorithm derived from the em algorithm for the mixture problem, Comp. Statis. Quaterly 2 (1985) 73–82.
[21] E. Kuhn, M. Lavielle, Coupling a stochastic approximation version of EM with an MCMC procedure, ESAIM Probab. Stat. 8 (2004) 115–131 (electronic).
[22] C. Andrieu, É. Moulines, P. Priouret, Stability of stochastic approximation under verifiable conditions, SIAM J. Control Optim. 44 (1) (2005) 283–312 (electronic).
[23] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Royal. Statist. Soc B 58, No. 1 (1996) 267–288.
[24] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics.
[25] D. Li, X. Sun, Nonlinear Integer Programming, 2006.
[26] A constrained em algorithm for independent component analysis, Neural Computing 13 (2001) 677–689.
[27] A. Hyvarinen, E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Computation.
[28] D. Grimes, A. Shon, R. Rao, Probabilistic bilinear models for appearance-based vision, Proc of the Ninth IEEE International Conference on Computer Vision (ICCV'03) 2 (2003) 1478–1486.
[29] K. Brandt Petersen, O. Winther, The em algorithm in independent component analysis, Proc of the ICASSP conference (2005) 169–172.
[30] M. I. Miller, A. Trouvé, L. Younes, On the metrics and Euler-Lagrange equations of computational anatomy, Annual Review of biomedical Engineering 4.
[31] M. I. Miller, A. Trouvé, L. Younes, Geodesic shooting for computational anatomy, Journal of Mathematical Imaging and Vision 24 (2) (2006) 209–228. doi:http://dx.doi.org/10.1007/s10851-005-3624-0.
[32] A. Trouvé, Diffeomorphism groups and pattern matching in image analysis, Int. J. of Comp. Vis. 28 (3) (1998) 213–221.
[33] A. Trouvé, L. Younes, Local geometry of deformable templates, Tech. rep., Université Paris 13 (2002).
[34] S. Allassonnière, E. Kuhn, Stochastic algorithm for bayesian mixture effect template estimation, In press in ESAIM Probab.Stat.